|  |  |  |
| --- | --- | --- |
|  |  |  |

**TÀI LIỆU ĐẶC TẢ API  
NHÓM 01**

|  |  |
| --- | --- |
| **Mã hợp đồng** | S.ARVR.01 |
| **Tên dự án** | Tích hợp AR VR vào quy trình mua sắm trên ứng dụng Shopee |
| **Phân hệ** | 1.0 |
| **Mã tài liệu** | S.ARVR.01 |
| **Phiên bản tài liệu** | 1.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TRANG KÝ** | | | | |
| **I. SOẠN THẢO** | | | | |
| **TT** | **Họ tên** | **Chức vụ** | **Ngày ký** | **Chữ ký** |
| 1. | Trương Anh Kiệt | Business Analyst | 14/10/2025 | Kiet |
| 2. | Trần Thanh Phát | Business Analyst | 16/10/2025 | Phat |
| 3. | Trịnh Mộng Hân | Business Analyst | 16/10/2025 | Han |
| 4. |  |  |  |  |
| 5. |  |  |  |  |
| 6. |  |  |  |  |
| **II. KIỂM SOÁT** | | | | |
| **TT** | **Họ tên** | **Chức vụ** | **Ngày ký** | **Chữ ký** |
| 1. | Trương Anh Kiệt | Business Analyst | 14/10/2025 | Kiet |
| 2. | Trần Thanh Phát | Business Analyst | 16/10/2025 | Phat |
| 3. | Trịnh Mộng Hân | Business Analyst | 16/10/2025 | Han |
| 4. |  |  |  |  |
| 5. |  |  |  |  |
| 6. |  |  |  |  |
| 7. |  |  |  |  |
| **III. PHÊ DUYỆT** | | | | |
| **TT** | **Họ tên** | **Chức vụ** | **Ngày ký** | **Chữ ký** |
| 1. | Trương Anh Kiệt | Business Analyst | 14/10/2025 | Kiet |
| 2. | Trần Thanh Phát | Business Analyst | 16/10/2025 | Phat |
| 3. | Trịnh Mộng Hân | Business Analyst | 16/10/2025 | Han |
| 4. |  |  |  |  |

**BẢNG GHI NHẬN THAY ĐỔI PHIÊN BẢN**

| **Ngày thay đổi** | **Phiên bản** | **Lý do** | **Mô tả thay đổi** |
| --- | --- | --- | --- |
| 14/10/2025 | 1 | Khởi tạo | Khởi tạo tài liệu đặc tả API |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**MỤC LỤC**

[I. Giới thiệu 7](#_Toc211422713)

[1.1 Mục tiêu 7](#_Toc211422714)

[1.2 Phạm vi 7](#_Toc211422715)

[II. Kiến trúc tích hợp 8](#_Toc211422716)

[2.1 Sơ đồ kiến trúc 8](#_Toc211422717)

[2.3 UML Sequences flow 8](#_Toc211422718)

[*III.* Chi tiết tích hợp (Integration & Endpoint) 9](#_Toc211422719)

[3.1 Danh sách các Service/ API 9](#_Toc211422720)

[3.2 Các API Endpoint của các hệ thống liên quan 10](#_Toc211422721)

[1. API: GET /api/products/{id}/3d-model 10](#_Toc211422722)

[2. API:POST /api/products/3d-model/upload 12](#_Toc211422723)

[3. API: GET /api/products/{id}/ar-preview 14](#_Toc211422724)

[4. API: POST /api/ai/face-detect 16](#_Toc211422725)

[5. API: POST /api/ai/body-measure 18](#_Toc211422726)

[6. API: POST /api/ai/room-scan 20](#_Toc211422727)

[7. API: POST /api/ai/recommendation 22](#_Toc211422728)

[8. API: POST /api/ar/session/start 24](#_Toc211422729)

[9. API: POST /api/analytics/ar-interaction 26](#_Toc211422730)

[10. API: GET /api/analytics/conversion-rate 28](#_Toc211422731)

**CÁC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Giải thích nghĩa** |
|  |  |  |
|  |  |  |
|  |  |  |

# Giới thiệu

## 1.1 Mục tiêu

Tài liệu này mô tả các API chính được sử dụng trong hệ thống Ứng dụng AR/VR tích hợp với sàn Thương mại Điện tử Shopee. Mục tiêu của các API này là cung cấp khả năng hiển thị sản phẩm 3D, nhận diện khuôn mặt/cơ thể/không gian bằng AI, và theo dõi hành vi người dùng trong môi trường AR/VR. Các API được thiết kế theo chuẩn RESTful, hỗ trợ JSON, có khả năng mở rộng và tích hợp linh hoạt giữa frontend (WebXR/Unity) và backend Shopee.

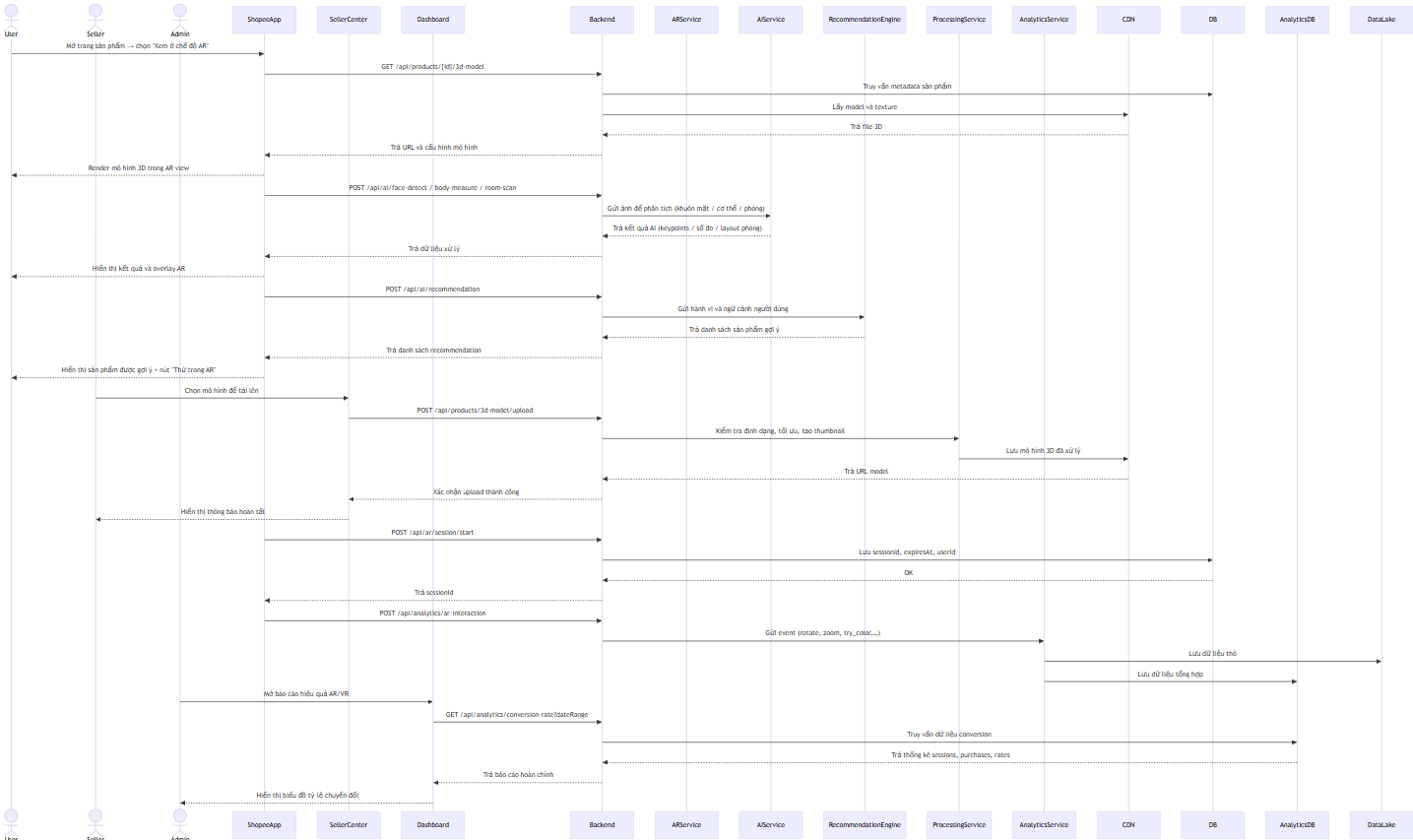
## 1.2 Phạm vi

Phạm vi của tài liệu bao gồm các API chính được gọi từ hệ thống Shopee để tích hợp và giao tiếp với các dịch vụ AR/VR, AI và hệ thống backend.  
 Tài liệu mô tả quy trình xử lý dữ liệu, luồng tương tác giữa các module và các yêu cầu kỹ thuật cần thiết để vận hành chức năng AR/VR trên nền tảng Shopee.

# Kiến trúc tích hợp

## 2.1 Sơ đồ kiến trúc

## 2.3 UML Sequences flow



Sơ đồ tổng quy trình tích hợp API vào ứng dụng Shopee

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhóm** | **Tên Actor / Module** | **Vai trò chính** | **Tham gia trong API nào** |
| 👤 **Actors (Người tương tác)** | | | |
| User | Người dùng cuối (khách hàng Shopee) | Trải nghiệm AR/VR, thử sản phẩm, xem nội thất | 1, 3, 4, 5, 6, 7, 8, 9 |
| Seller | Nhà bán hàng trên Shopee | Tải lên mô hình 3D, quản lý sản phẩm | 2 |
| Admin | Nhân viên Shopee quản trị / phân tích | Xem báo cáo, theo dõi conversion | 10 |
| 👨‍💻 **Ứng dụng & giao diện người dùng** | | | |
| Shopee App / WebXR | Ứng dụng chính cho người dùng; chứa giao diện AR/VR và API client | Gọi và nhận dữ liệu từ Backend, hiển thị AR/VR | 1–9 |
| Seller Center | Giao diện web cho nhà bán hàng, cho phép upload model | Upload model và nhận phản hồi | 2 |
| Dashboard (Admin Portal) | Ứng dụng quản trị Shopee nội bộ | Xem số liệu, conversion rate, KPI AR/VR | 10 |
| ⚙️ **Backend & Services nội bộ** | | | |
| Shopee Backend | API Gateway, xử lý logic nghiệp vụ, xác thực, lưu trữ metadata, điều phối các services | Là trung tâm của toàn bộ 10 API | 1–10 |
| AI Service | Xử lý AI: nhận diện khuôn mặt, cơ thể, không gian, gợi ý | Phục vụ các API 4–7 | 4, 5, 6, 7 |
| AR Service | Xử lý cấu hình hiển thị AR, ánh sáng, tương tác 3D | Trả dữ liệu scale, light profile | 3 |
| Recommendation Engine | Mô hình Machine Learning gợi ý sản phẩm | Gợi ý sản phẩm theo hành vi và ngữ cảnh | 7 |
| Analytics Service | Thu nhận và xử lý event tương tác AR/VR | Ghi dữ liệu, sinh báo cáo | 9, 10 |
| Processing Service | Xử lý mô hình 3D khi upload (optimize, generate thumbnail) | Dành cho API upload model | 2 |
| CDN (Content Delivery Network) | Lưu trữ mô hình, texture, ảnh preview | Phục vụ phân phối file nhanh chóng | 1, 2 |
| DB (Database) | Lưu thông tin sản phẩm, metadata, cấu hình AR, mapping size chart | Truy vấn / lưu trữ dữ liệu nền | 1–10 |
| Analytics DB | Lưu dữ liệu tổng hợp, chuyển đổi, tỷ lệ conversion | Trả báo cáo | 10 |
| Data Lake | Lưu dữ liệu thô từ Analytics Service | Nguồn dữ liệu phân tích sâu | 9, 10 |

Bảng Modules Actors chính

# Chi tiết tích hợp (Integration & Endpoint)

## 3.1 Danh sách các Service/ API

|  |  |  |
| --- | --- | --- |
| **STT** | **Endpoint** | **Mô tả** |
| 1 | GET /api/products/{id}/3d-model | Lấy mô hình 3D sản phẩm để hiển thị trong AR/VR |
| 2 | POST /api/products/3d-model/upload | Upload mô hình 3D sản phẩm |
| 3 | GET /api/products/{id}/ar-preview | Lấy thông tin hiển thị sản phẩm AR |
| 4 | POST /api/ai/face-detect | Nhận diện khuôn mặt phục vụ thử mỹ phẩm |
| 5 | POST /api/ai/body-measure | Phân tích cơ thể người dùng để thử trang phục |
| 6 | POST /api/ai/room-scan | Quét không gian phòng để bố trí nội thất ảo |
| 7 | POST /api/ai/recommendation | Gợi ý sản phẩm phù hợp từ dữ liệu AI |
| 8 | POST /api/ar/session/start | Khởi tạo phiên làm việc AR |
| 9 | POST /api/analytics/ar-interaction | Ghi nhận hành vi tương tác AR/VR |
| 10 | GET /api/analytics/conversion-rate | Truy xuất thống kê chuyển đổi mua hàng AR/VR |

## 3.2 Các API Endpoint của các hệ thống liên quan

### 1. API: GET /api/products/{id}/3d-model

**1.1. Business Purpose**

Cho phép client (Shopee App / WebXR / Unity) truy xuất mô hình 3D & metadata của sản phẩm để hiển thị trong AR/VR. Hỗ trợ trải nghiệm “thử trước khi mua” cho các ngành nội thất, thời trang, phụ kiện, mỹ phẩm.

**1.2. Detailed Flow**

1. Client gửi GET /api/products/{id}/3d-model cùng token.
2. Backend xác thực token và quyền truy cập.
3. Backend truy vấn DB sản phẩm để kiểm tra tồn tại và trạng thái 3D model.
4. Nếu model có sẵn, Backend gọi CDN để lấy modelUrl/optimizedModelUrl/thumbnailUrl.
5. Backend trả về metadata (kích thước, định dạng, kích thước tập tin, ngày cập nhật).
6. Client tải mô hình (hoặc bản tối ưu) và render bằng WebXR/Unity.
7. Nếu người dùng tương tác, client gọi API analytics tương ứng.

**1.3. Request**

**Method:** GET  
**Endpoint:** /api/products/{productId}/3d-model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Type** | **Required** | **Location** | **Description** |
| productId | string | Yes | Path | ID sản phẩm |
| format | string | No | Query | Yêu cầu định dạng (glb, usdz, fbx) |
| quality | string | No | Query | high / medium / low (ảnh hưởng optimizedModelUrl) |
| Accept-Language | string | No | Header | Ngôn ngữ (vn/en...) |

**Headers:** Authorization: Bearer <token>

**Example Request**GET /api/products/10345/3d-model?format=glb&quality=medium

Authorization: Bearer eyJhbGci...

Accept-Language: vn

**1.4. Repsonse (200)**

{

"status": "success",

"data": {

"productId": "10345",

"productName": "Modern Wooden Chair",

"category": "Furniture",

"modelUrl": "https://cdn.shopee.vn/models/10345.glb",

"optimizedModelUrl": "https://cdn.shopee.vn/models/10345\_opt\_low.glb",

"textureUrl": "https://cdn.shopee.vn/textures/10345\_diffuse.png",

"thumbnailUrl": "https://cdn.shopee.vn/previews/10345-thumb.jpg",

"fileSize": 13421772,

"format": "glb",

"lastUpdated": "2025-10-14T09:45:00Z",

"cacheTtl": 86400

}

}

**Response fields (giải thích)**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Mô tả** |
| modelUrl | string (URL) | Link tới file 3D chính |
| optimizedModelUrl | string (URL) | Link tới bản tối ưu (nếu có) |
| textureUrl | string | Link texture / PBR maps |
| thumbnailUrl | string | Ảnh xem trước |
| fileSize | int | Kích thước byte |
| format | string | Định dạng file |
| cacheTtl | int | TTL cho cache (giây) |

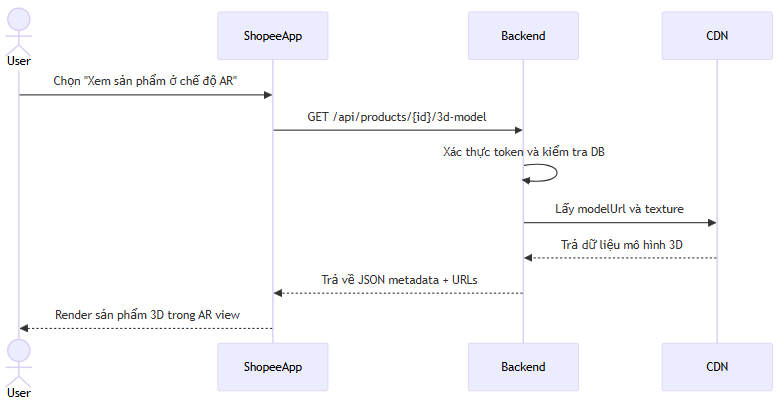
**Errors (ví dụ):**

* 401 Unauthorized — token thiếu/không hợp lệ
* 404 Not Found — sản phẩm hoặc model không tồn tại
* 413 Payload Too Large — kích thước model vượt quy định (trong trường hợp upload từ client)
* 500 Internal Server Error — lỗi backend/CDN

**1.5. Notes:**

* Hỗ trợ định dạng: .glb, .usdz, .fbx.
* File tối đa khuyến nghị: **50 MB**; backend có cơ chế trả optimizedModelUrl cho chất lượng thấp hơn.
* Metadata cache 24h theo cacheTtl.
* Yêu cầu kiểm tra quyền truy cập (seller-only model có thể private).
* Audit log: mọi request trả về 3D model cần ghi vào audit để phục vụ phân tích.

**1.6. Sequence Diagram:**

****

### 2. API:POST /api/products/3d-model/upload

**2.1. Business Purpose**

Cho phép **Seller** hoặc internal operator upload/replace 3D model cho sản phẩm. Cần hỗ trợ validate file, tự động tối ưu, tạo preview thumbnail và cập nhật metadata để client có thể hiển thị.

**2.2. Detailed Flow**

1. Seller gửi file 3D (multipart/form-data hoặc base64) qua SellerCenter.
2. Backend xác thực seller, kiểm tra quyền quản lý productId.
3. Backend chạy validation: định dạng hợp lệ, polygon count, kích thước.
4. Nếu vượt ngưỡng, trả lỗi kèm gợi ý (optimize, compress).
5. Nếu hợp lệ, Backend lưu tạm, gọi Cloud Processing Service để chuyển đổi/optimize/validate thêm (generate LODs, thumbnails).
6. Sau khi xử lý, upload file lên CDN, cập nhật DB metadata (modelUrl, sizes, LODs).
7. Trả về kết quả upload: URLs, status, warnings (nếu có).

**2.3. Request**

**Method:** POST  
**Endpoint:** /api/products/3d-model/upload **Content-Type:** multipart/form-data hoặc application/json (base64)

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| productId | string | Yes | ID sản phẩm |
| file | file | Yes | File 3D (.glb/.fbx/.usdz) |
| fileName | string | No | Tên file |
| generatePreview | boolean | No (default true) | Yêu cầu generate thumbnail/preview |
| visibility | string | No (public/private) | Quyền truy cập model |

**Example (multipart)**

POST /api/products/3d-model/upload

Authorization: Bearer <token>

Form-data:

- productId = 10345

- file = (chair\_model.glb)

- visibility = public

**2.4. Response (201 Created)**

{

"status": "success",

"message": "Model uploaded and processed.",

"data": {

"productId": "10345",

"modelUrl": "https://cdn.shopee.vn/models/10345.glb",

"optimizedModelUrl": "https://cdn.shopee.vn/models/10345\_opt\_low.glb",

"previewUrl": "https://cdn.shopee.vn/previews/10345-thumb.jpg",

"processingStatus": "completed",

"warnings": []

}

}

**Response fields (giải thích)**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Mô tả** |
| processingStatus | string | queued / processing / completed / failed |
| warnings | array | Các cảnh báo (ví dụ: polygon count cao) |

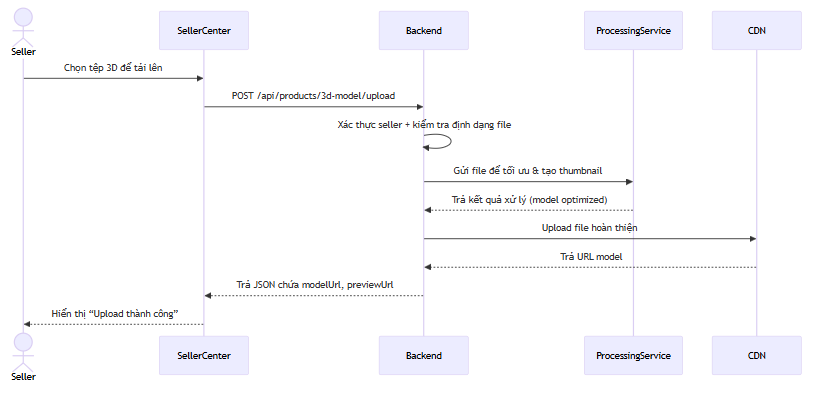
**Errors (ví dụ):**

* 400 Bad Request — thiếu productId hoặc file định dạng không hỗ trợ
* 401 Unauthorized — không có quyền upload
* 422 Unprocessable Entity — file hợp lệ nhưng vượt giới hạn polygons/texture size (kèm suggestions)
* 500 — lỗi xử lý background

**2.5. Notes:**

* Giới hạn file upload: <= 100MB (tùy policy), polygon limit: e.g. <= 500k faces.
* Hệ thống khuyến nghị upload .glb cho web, .usdz cho iOS AR Quick Look.
* Sau upload, model trải qua pipeline: validation → LOD generation → thumbnail → CDN.
* Upload có thể trả processingStatus=queued; client nên poll hoặc subscribe webhooks để cập nhật.

**2.6. Sequence Diagram:**



### 3. API: GET /api/products/{id}/ar-preview

**3.1. Business Purpose**

Trả về **cấu hình hiển thị AR** cho sản phẩm: scale mặc định, offset để đặt lên mặt phẳng, ánh sáng đề xuất, anchor points và collision box. Giúp client render đúng tỷ lệ và tương thích với môi trường người dùng.

**3.2. Detailed Flow**

1. Client gọi API để lấy AR configuration cho productId.
2. Backend lấy thông tin từ DB: anchorPoints, defaultScale, rotation, preferredLightingProfile, collisionBox.
3. Nếu sản phẩm có nhiều variants, trả config theo variantId.
4. Backend trả metadata plus optional materialOverrides (khi sản phẩm có nhiều materials).

**3.3. Request**

**Method:** GET  
**Endpoint:** /api/products/{productId}/ar-preview

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **Required** | **Description** |
| productId | string | Yes | ID sản phẩm |
| variantId | string | No | variant cụ thể |
| environment | string | No | indoor / outdoor để điều chỉnh light |

**Example Request**GET /api/products/10345/ar-preview?environment=indoor

Authorization: Bearer <token>

**3.4. Repsonse (200)**

{

"status": "success",

"data": {

"productId": "10345",

"defaultScale": 1.0,

"rotation": { "x": 0, "y": 90, "z": 0 },

"anchorPoints": [

{ "name": "seat", "x": 0.0, "y": 0.4, "z": 0.0 }

],

"collisionBox": { "min": [-0.6, 0, -0.6], "max": [0.6, 1.2, 0.6] },

"preferredLightingProfile": "softWarm",

"materialOverrides": [

{ "materialName": "leather", "albedo": "#b5651d", "roughness": 0.4 }

]

}

}

**Fields table:**

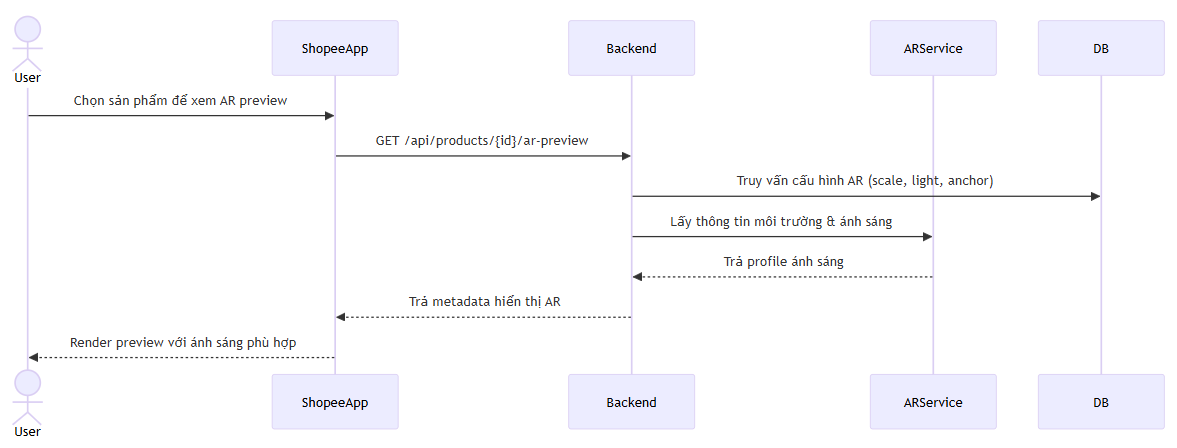
|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Mô tả** |
| anchorPoints | array | Dùng để gắn phụ kiện hoặc vị trí đo trong AR |
| collisionBox | object | Giúp physics/placement tránh đè chồng vật thật |
| preferredLightingProfile | string | Tên profile ánh sáng (mapped trên ARService) |

**Errors (ví dụ):** 401, 404 (không có config), 500

**3.5. Notes:**

* AR client nên fallback nếu materialOverrides rỗng.
* defaultScale tương ứng với unit mặc định (meter). Tài liệu phải nêu rõ unit.
* Nếu product có âm lượng lớn (room-scale), backend có thể trả thêm placementHints.

**3.6. Sequence Diagram:**



### 4. API: POST /api/ai/face-detect

**4.1. Business Purpose**

Xử lý nhận diện khuôn mặt, trả về keypoints (mắt, mũi, môi), skinTone, facial landmarks để overlay mỹ phẩm ảo, chọn màu phù hợp và kiểm tra scale/fit cho kính, mũi nón.

**4.2. Detailed Flow**

1. Client thu ảnh camera (single frame hoặc stream snapshot) và gửi base64 hoặc multipart.
2. Backend xác thực, chuyển ảnh đến AIService (face-detection model).
3. AIService trả keypoints, bounding box, confidence, skinTone estimation, headPose.
4. Backend có thể enrich bằng lookup (ví dụ mapping skinTone -> palette recommended).
5. Trả về client; client áp dụng overlay real-time.

**4.3. Request**

**Method:** POST  
**Endpoint:** /api/ai/face-detect **Content-Type:** application/json (base64) hoặc multipart/form-data

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| imageBase64 | string | Yes | Base64 encoded image (RGB) |
| detectLandmarks | boolean | No (default true) | Có cần keypoints chi tiết không |
| returnSkinTone | boolean | No | Có trả kiểu da không |
| maxFaces | int | No | Số mặt tối đa cần detect (default 1) |

**Example**

{

"imageBase64": "<base64-image>",

"detectLandmarks": true,

"returnSkinTone": true

}

**4.4. Response (200)**

{

"status": "success",

"data": {

"faces": [

{

"faceId": "F-987",

"boundingBox": { "x": 100, "y": 80, "w": 200, "h": 220 },

"landmarks": {

"leftEye": [150, 120], "rightEye": [250, 118], "noseTip": [200, 160], "mouth": [200, 210]

},

"headPose": { "yaw": 2.5, "pitch": -3.0, "roll": 0.1 },

"skinTone": "warm-medium",

"confidence": 0.98

}

],

"processingTimeMs": 42

}

}

**Response fields (giải thích)**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Mô tả** |
| landmarks | object | Coordinates in image pixel space |
| headPose | object | Yaw/pitch/roll — giúp align overlay |

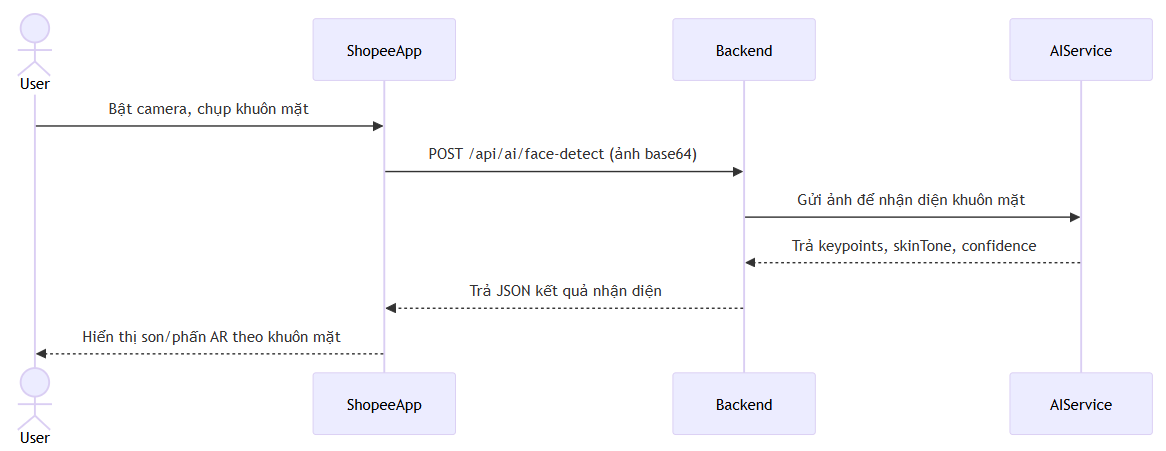
**Errors (ví dụ):**

* 400 Invalid image
* 422 No face detected (kèm message)
* 429 Rate limit if nhiều request từ 1 device

**4.5. Notes:**

* Không lưu ảnh theo mặc định; nếu lưu cần consent người dùng.
* Thời gian xử lý mục tiêu <100ms cho realtime (tùy infra).
* Trả về tọa độ theo hệ toạ độ ảnh (client phải chuyển sang viewport space).
* Bảo mật: image payload phải truyền qua HTTPS và có token.

**4.6. Sequence Diagram:**



### 5. API: POST /api/ai/body-measure

**5.1. Business Purpose**

Dự đoán số đo cơ thể (height, chest, waist, hips) từ ảnh hoặc video, trả về recommendedSize mapping theo brand size-chart, giảm rủi ro chọn sai size khi thử đồ ảo.

**5.2. Detailed Flow**

1. Client yêu cầu user chụp ảnh toàn thân (tối thiểu 2 góc: front & side) hoặc sử dụng depth data (nếu có).
2. Ảnh gửi tới backend rồi forward đến AIService (pose estimation + 3D reconstruction).
3. AIService trả số đo ± error margin.
4. Backend áp dụng brand-specific size chart (từ DB) để map sang recommendedSize (S/M/L...).
5. Trả về client kèm confidence và errorMargin.

**5.3. Request**

**Method:** POST  
**Endpoint:** /api/ai/body-measure  
**Content-Type:** application/json

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| images | array[string] | Yes | Array of base64 images (front, optional side) |
| heightDeclared | int | No | Chiều cao do user nhập (cm) giúp tăng chính xác |
| gender | string | No | male / female |
| brandId | string | No | Để map size chart specific brand |

**Example:**

{

"images": ["<base64-front>", "<base64-side>"],

"heightDeclared": 165,

"gender": "female",

"brandId": "BRAND-001"

}

**5.4. Reponse (200)**

{

"status": "success",

"data": {

"heightCm": 164.5,

"chestCm": 86,

"waistCm": 64,

"hipsCm": 90,

"recommendedSize": "M",

"confidence": 0.87,

"errorMargin": { "height": 1.5, "chest": 2.0 }

}

}

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Mô tả** |
| errorMargin | object | Sai số ước tính ± cm |

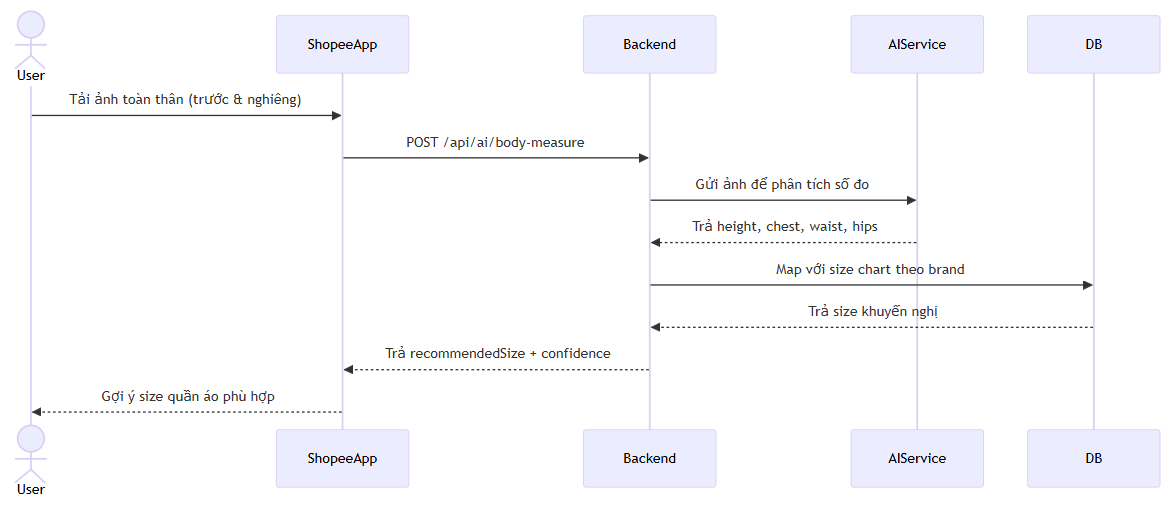
**Errors**

* 400 invalid images
* 422 insufficient images (yêu cầu ít nhất 1 frontal + optional side)

**5.5. Notes:**

* Yêu cầu hướng dẫn user chụp ảnh đúng (full body, neutral pose).
* Nếu user cung cấp heightDeclared, AI dùng như priors.
* Không ghi nhận ảnh theo mặc định; cần consent nếu lưu.
* Với brand-specific sizing, cần cập nhật size chart định kỳ.

**5.6. Sequence Diagram:**

****

### 6. API: POST /api/ai/room-scan

**6.1. Business Purpose**

Phân tích ảnh/quét phòng để trả về kích thước phòng, mặt phẳng đặt, vật thể đã có, ánh sáng; phục vụ tính năng đặt nội thất ảo đúng tỷ lệ (IKEA-like).

**6.2. Detailed Flow**

1. Client hướng dẫn user quét phòng (pan video hoặc nhiều ảnh).
2. Client gửi frames hoặc point cloud (nếu có depth) lên backend.
3. AIService chạy SLAM / semantic segmentation -> trả roomDimensions, floorPlane, detectedObjects (sofa, table...).
4. Backend dùng kết quả để gợi ý sản phẩm phù hợp hoặc auto-place items.

**6.3. Request**

**Method:** POST  
**Endpoint:** /api/ai/room-scan

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| frames | array[string] | Yes | Base64 frames hoặc video snippet |
| useDepth | boolean | No | Nếu true, payload kèm depth/pointcloud |
| preferredStyle | string | No | minimalist, scandinavian... |

**Example:**

{

"frames": ["<base64-frame1>", "<base64-frame2>", "..."],

"preferredStyle": "minimalist"

}

**6.4. Reponse (200)**

{

"status": "success",

"data": {

"roomDimensions": { "widthM": 4.0, "depthM": 5.2, "heightM": 2.8 },

"floorPlane": { "normal": [0,1,0], "position": [0,0,0] },

"lighting": { "type": "warm", "luxEstimate": 220 },

"detectedObjects": [

{ "type": "sofa", "bbox": [1.2,0,2.3,0.8], "confidence": 0.95 },

{ "type": "lamp", "bbox": [3.4,0,1.1,0.4], "confidence": 0.88 }

],

"suggestedPlacements": [

{ "productId": "COF-123", "position": [1.0,0,2.0], "rotationY": 180 }

]

}

}

|  |  |
| --- | --- |
| **Field** | **Mô tả** |
| suggestedPlacements | Gợi ý item + vị trí để client auto-place trong AR |

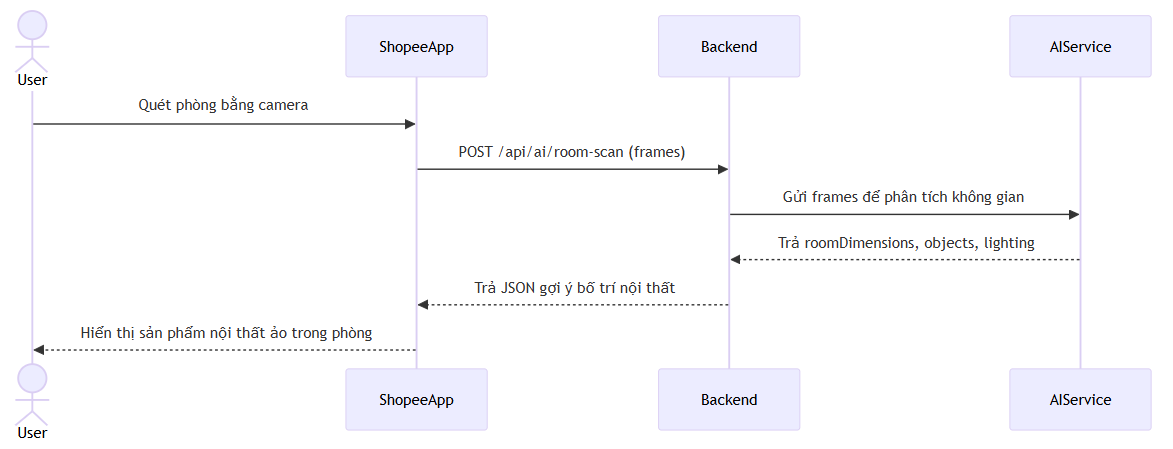
**Errors**

* 400 invalid frames
* 422 insufficient coverage (cần quét đầy đủ phòng)

**6.5. Notes:**

* Quét yêu cầu đủ coverage (front + sides); hướng dẫn UX bắt buộc.
* Nếu depth unavailable, accuracy thấp hơn.
* Privacy: frames không lưu trừ khi consent.

**6.6. Sequence Diagram:**



### 7. API: POST /api/ai/recommendation

**7.1. Business Purpose**

Dựa trên hành vi AR/VR (colors tried, items placed, face/body traits), kết hợp historical purchase data để trả về **personalized product recommendations** ưu tiên những item có hỗ trợ AR/VR.

**7.2. Detailed Flow**

1. Client gửi context: userId, recentInteractions, device, currentScene (e.g., tried lipstick red).
2. Backend môi trường enrich user profile (loyalty tier, demographics).
3. Backend gọi Recommendation Engine (ML model).
4. Model trả ranked list with scores & explainability tokens (ví dụ: because you tried red lipstick).
5. Backend trả list cho client; client có nút “Try in AR” cho mỗi item.

**7.3. Request**

**Method:** POST  
**Endpoint:** /api/ai/recommendation

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| userId | string | Yes | ID người dùng |
| context | string | Yes | beauty / furniture / fashion |
| recentInteractions | array[string] | No | Action ids: lipstick\_red, placed\_table |
| limit | int | No (default 10) | Số item trả về |

**Example:**

{

"userId": "U1234",

"context": "beauty",

"recentInteractions": ["lipstick\_red", "eyeshadow\_pink"],

"limit": 5

}

**7.4. Reponse (200)**

{

"status": "success",

"data": {

"recommendations": [

{

"productId": "SP2024",

"name": "Matte Red Lipstick",

"score": 0.92,

"reason": "Matched color you tried",

"supportsAr": true

},

{

"productId": "SP3011",

"name": "Soft Pink Blush",

"score": 0.87,

"reason": "Completed look with eyeshadow",

"supportsAr": true

}

],

"modelVersion": "rec-v2.3",

"processingTimeMs": 85

}

}

|  |  |
| --- | --- |
| **Field** | **Mô tả** |
| score | Ranking score 0..1 |
| reason | Short explainability string |

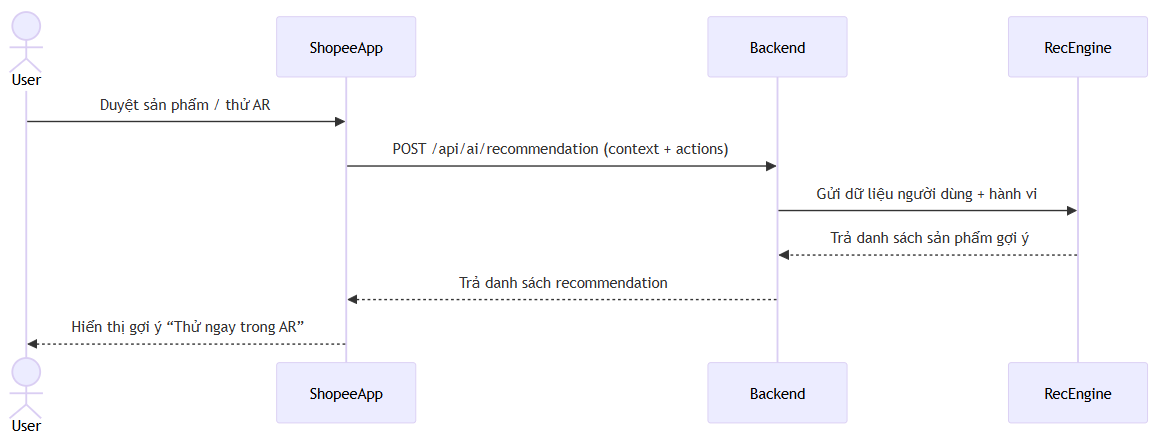
**Errors**

* 400 missing userId/context
* 503 Recommendation service unavailable

**7.5. Notes:**

* Thứ tự đề xuất ưu tiên items supportsAr=true để nâng UX AR.
* Model versioning rất quan trọng (field modelVersion).
* Phải giới hạn rate per user để tránh quá tải model.

**7.6. Sequence Diagram:**



### 8. API: POST /api/ar/session/start

**8.1. Business Purpose**

Tạo session cho trải nghiệm AR/VR nhằm quản lý state, quotas, A/B test variants, gắn sessionId cho analytics events và hỗ trợ resume.

**8.2. Detailed Flow**

1. Client gửi request tạo session (userId, device, mode).
2. Backend xác thực, tạo session record (sessionId, expiresAt, features flags).
3. Backend trả sessionId, policies (maxEventsPerMinute), and recommended quality settings.
4. Client dùng sessionId cho các request analytics / interaction.

**8.3. Request**

**Method:** POST  
**Endpoint:** /api/ar/session/start

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| userId | string | Yes | ID người dùng |
| deviceInfo | object | No | device model, os, capabilities |
| mode | string | Yes | AR / VR |
| experimentId | string | No | nếu tham gia A/B test |

**Example:**

{

"userId": "U1234",

"deviceInfo": {"model":"iPhone15","ramGb":6},

"mode": "AR",

"experimentId": "exp-ar-2025-10"

}

**8.4. Reponse (201)**

{

"status": "success",

"data": {

"sessionId": "AR-20251014-XYZ",

"createdAt": "2025-10-14T09:47:00Z",

"expiresIn": 3600,

"qualityProfile": "medium",

"maxEventsPerMinute": 120,

"features": { "cloudRendering": true, "enableHighRes": false }

}

}

|  |  |
| --- | --- |
| **Field** | **Mô tả** |
| qualityProfile | low/medium/high — guide client settings |
| features | flags feature toggles for session |

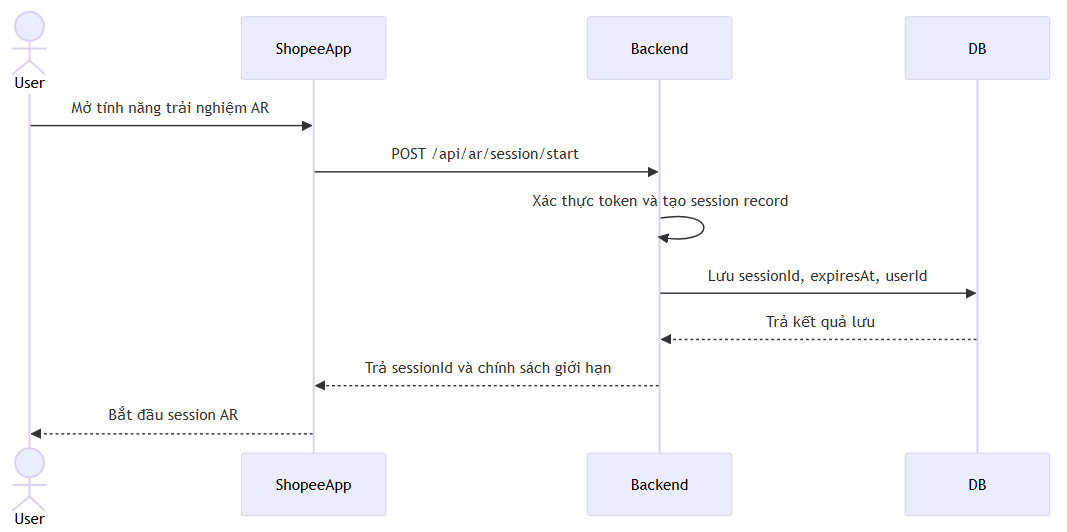
**Errors**

* 401 unauthorized
* 429 too many sessions from same user/device

**8.5. Notes:**

* Sessions expire để tránh resource leaks.
* Session dùng cho throttling analytics, billing (nếu áp dụng).
* Có thể hỗ trợ resume nếu sessionId còn valid.

**8.6. Sequence Diagram:**



### 9. API: POST /api/analytics/ar-interaction

**9.1. Business Purpose**

Ghi nhận mọi hành vi tương tác (event) trong phiên AR/VR để phân tích UX, conversion funnel, A/B testing, và cải thiện recommendations.

**9.2. Detailed Flow**

1. Client thu event (rotate, zoom, tryColor, placedItem, addToCart) kèm sessionId và gửi batch events hoặc single event.
2. Backend validate sessionId, enrich event (userId, timestamp), forward to Analytics Service / Kafka.
3. Analytics service persist raw events and compute near-real-time metrics (e.g., unique sessions, avg interactions).
4. Endpoint trả ack (accepted) để client có thể flush queue.

**9.3. Request**

**Method:** POST  
**Endpoint:** /api/analytics/ar-interaction

**Body (array events allowed)**

{

"sessionId": "AR-20251014-XYZ",

"events": [

{

"eventType": "rotate\_model",

"productId": "10345",

"properties": {"degrees": 90},

"timestamp": "2025-10-14T09:47:12Z"

},

{

"eventType": "try\_color",

"productId": "SP2024",

"properties": {"color":"#E57373"},

"timestamp": "2025-10-14T09:47:20Z"

}

]

}

**Event schema (short)**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Mô tả** |
| eventType | string | rotate\_model, zoom, try\_color, place\_item, add\_to\_cart |
| properties | object | thêm dữ liệu contextual |

**Example:**

**9.4. Reponse (202 Accepted)**

{

"status": "accepted",

"message": "2 events queued",

"ingestId": "ingest-20251014-0001"

}

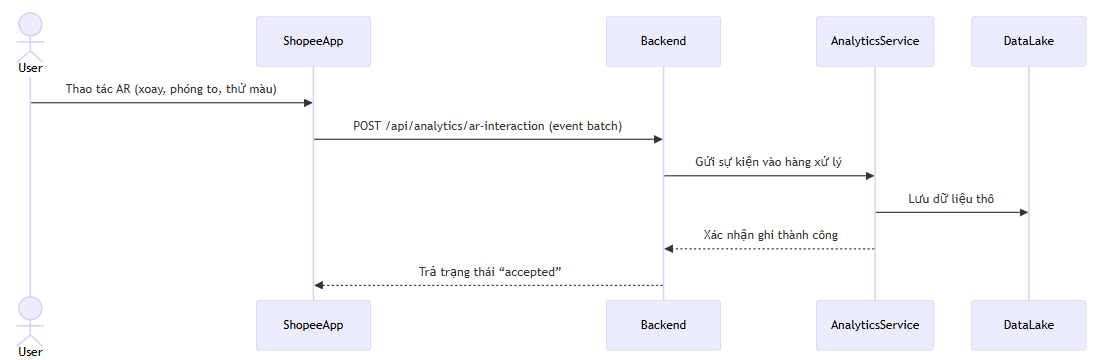
**Errors**

* 401 invalid session/token
* 429 ingestion rate limit
* 400 invalid schema

**9.5. Notes:**

* Hỗ trợ batch events (recommended) để giảm chattiness.
* Events có thể tuyến tính hoặc được buffering (client-side).
* SLA ingestion: target <200ms ack.
* Dữ liệu raw lưu vào Data Lake; aggregated metrics build daily/hourly.

**9.6. Sequence Diagram:**



### 10. API: GET /api/analytics/conversion-rate

**10.1. Business Purpose**

Truy xuất báo cáo tỉ lệ chuyển đổi từ trải nghiệm AR/VR (ví dụ: sessions with try -> addToCart -> purchase) để BA/marketing đánh giá ROI, hiệu quả công nghệ AR.

**10.2. Detailed Flow**

1. Admin/Dashboard gọi API với dateRange, groupBy (day/week/brand).
2. Backend query aggregated tables (pre-computed daily aggregates).
3. Nếu data chưa có, backend request to AnalyticsService to compute on-the-fly (slow path).
4. Trả về report gồm totalSessions, interactions, addToCart, purchases, conversionRate.

**10.3. Request**

**Method:** GET  
**Endpoint:** /api/analytics/conversion-rate

|  |  |  |  |
| --- | --- | --- | --- |
| **Param** | **Type** | **Required** | **Description** |
| startDate | date | Yes | YYYY-MM-DD |
| endDate | date | Yes | YYYY-MM-DD |
| groupBy | string | No (default day) | day/week/month |
| filter | string | No | e.g. category:Furniture |

**Example:**

GET /api/analytics/conversion-rate?startDate=2025-10-01&endDate=2025-10-14&groupBy=day&filter=category:Furniture

Authorization: Bearer <admin-token>

**10.4. Reponse (200)**

{

"status": "success",

"data": {

"dateRange": "2025-10-01 to 2025-10-14",

"totalSessions": 1200,

"totalInteractions": 9200,

"addToCart": 420,

"purchases": 348,

"conversionRate": 0.29,

"grouped": [

{ "date": "2025-10-01", "sessions": 80, "purchases": 20, "conversionRate": 0.25 },

...

]

}

}

|  |  |
| --- | --- |
| **Field** | **Mô tả** |
| conversionRate | purchases / totalSessions (float 0..1) |

**Errors**

* 401 unauthorized
* 400 invalid date range
* 503 analytics pipeline lagging

**10.5. Notes:**

* Thường dùng aggregated pre-computed tables for speed.
* Nếu filter requires granular join (user-level), may trigger on-the-fly compute (slow). System should inform response time in header X-Processing-Time.

**10.6. Sequence Diagram:**

